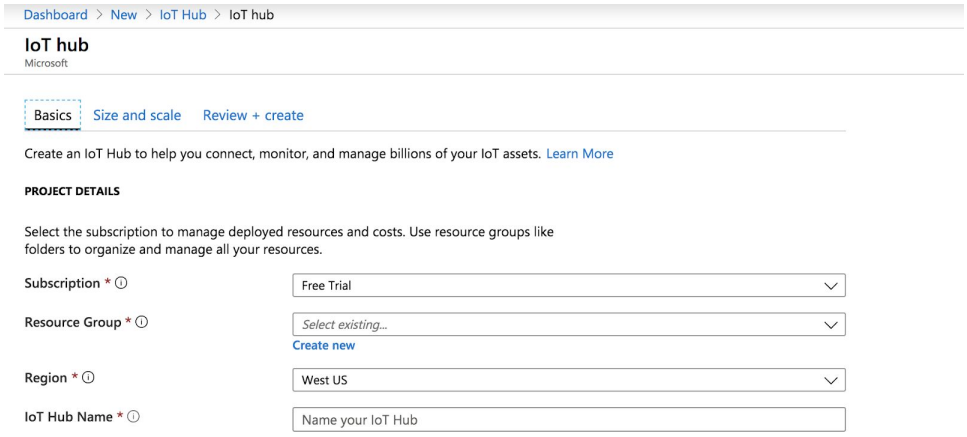
CSS532 HW2 Report

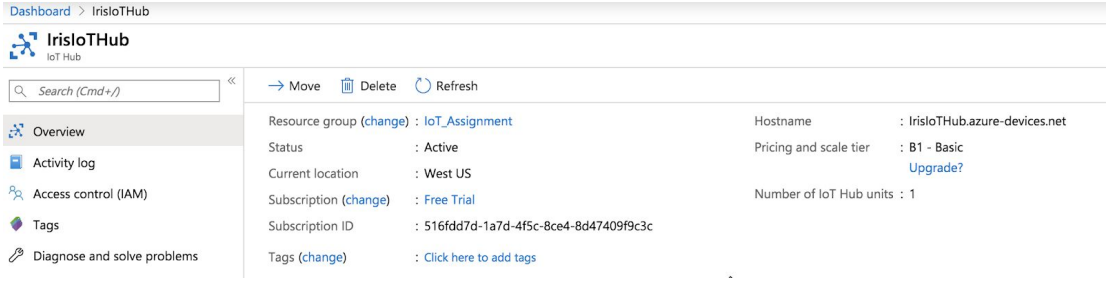
Name: Iris(Qiaoyu) Zheng

1 Feature Realization：

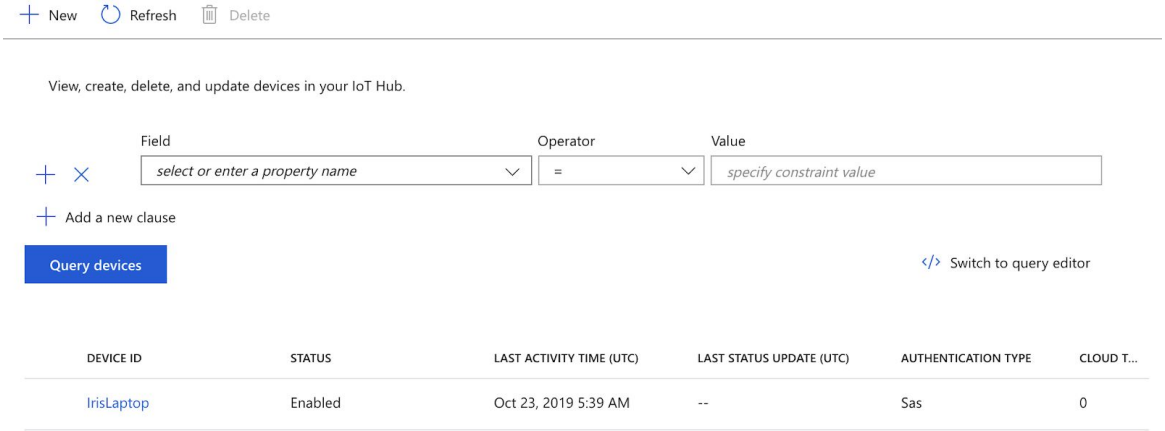
a). IoTConfigure your laptop as an IoT device, which can communicate with Azure IoT Hub

Create Azure IoT Hub





Add new devices onto IoT hub



Create a local python file, get connection string and paste the string to the python as a constant value.

Successfully connected the laptop to the Azure IoT Hub with Azure IoT SDK.

```
# The connection string for a device should never be stored in code. For the sake of simplicity we're using ar
conn_str = os.getenv("IOTHUB_DEVICE_CONNECTION_STRING")
device_id = "IrisLaptop"
host_name = "IrisIoTHub"
# The client object is used to interact with your Azure IoT hub.
device_client = IoTHubDeviceClient.create_from_connection_string("HostName=IrisIoTHub.azure-devices.net;Device

# Connect the client.
device_client.connect()
```

b). Configure Azure IoT Hub to receive messages from your device (laptop) and show the result in Azure IoT Hub console

Do the configuration in python to connect the laptop to the console and send the message.

Python code:

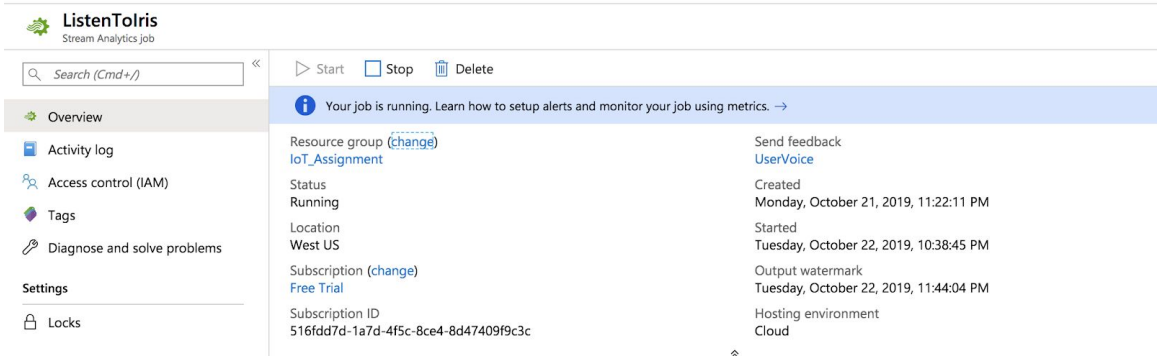
```
# The connection string for a device should never be stored in code. For the sake of simplicity we're using an
conn_str = os.getenv("IOTHUB_DEVICE_CONNECTION_STRING")
device_id = "IrisLaptop"
host_name = "IrisIoTHub"
# The client object is used to interact with your Azure IoT hub.
device_client = IoTHubDeviceClient.create_from_connection_string("HostName=IrisIoTHub.azure-devices.net;Device

# Connect the client.
device_client.connect()

message2 = {}
message2['data1'] = random.randrange(10)
message2['data2'] = random.randrange(10)
message2['data3'] = random.randrange(10)

messageJson2 = json.dumps(message2)
device_client.send_message(messageJson2)
time.sleep(1)

device_client.disconnect()
```

Monitor the message received on the IoT Console via iot hub monitor events:

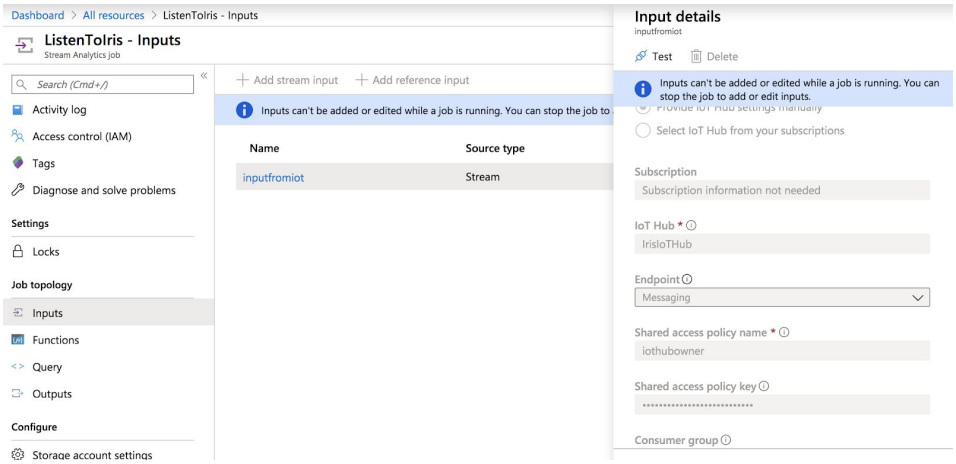
```
(base) Iriss-MacBook-Pro:Assignment2 iriszheng$ az iot hub monitor-events --hub-
name IrisIoTHub --device-id IrisLaptop
Starting event monitor, filtering on device: IrisLaptop, use ctrl-c to stop...
{
    "event": {
        "origin": "IrisLaptop",
        "payload": {
            "data1": 8,
            "data2": 2,
            "data3": 3
        }
    }
}
```

c). Connect Azure Stream Analytics to your Azure IoT Hub to process received messages from devices and save the raw data (the data shall be abstracted from received messages) and the processed data (you should use Stream Analytics to process the data) into Azure Blob.

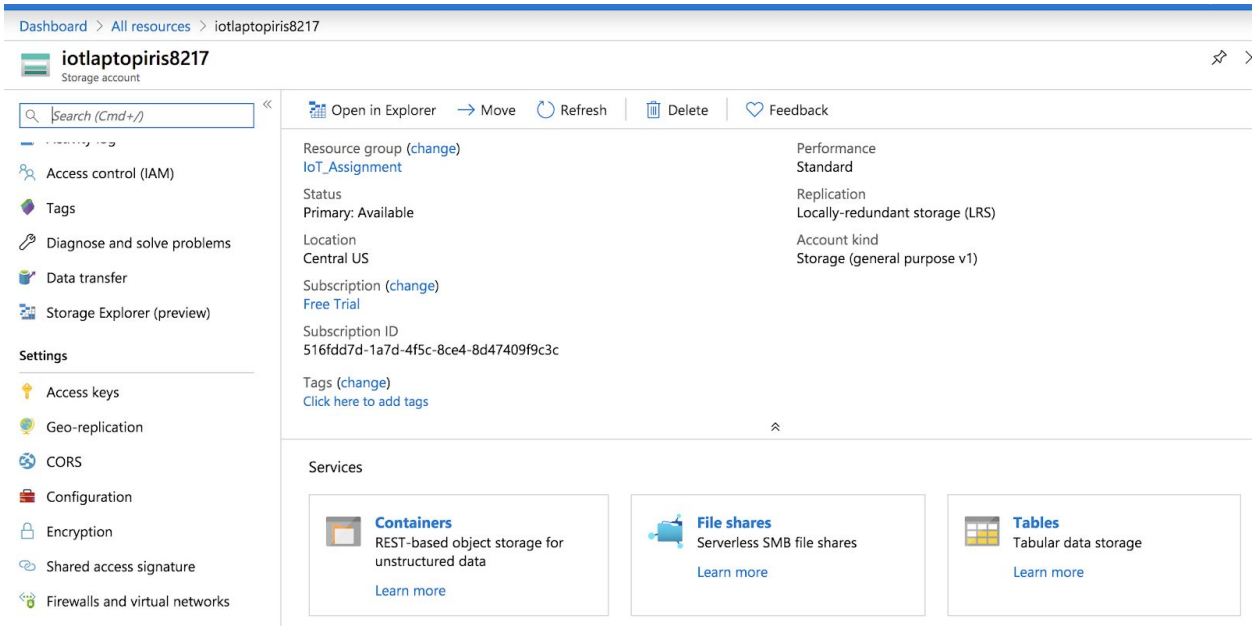
Create a stream analytic job



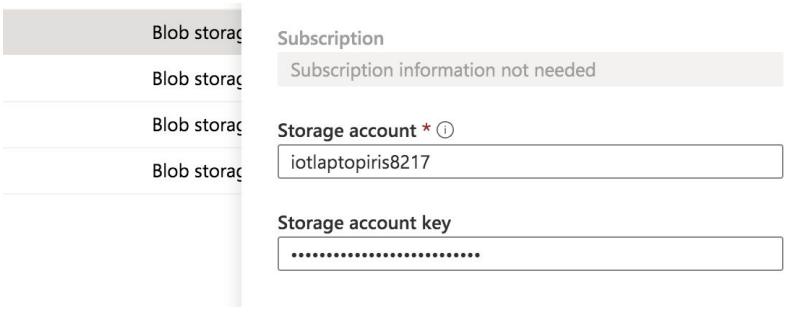
Add an input from IoT hub, set access policy name/key/ consumer group. Etc.



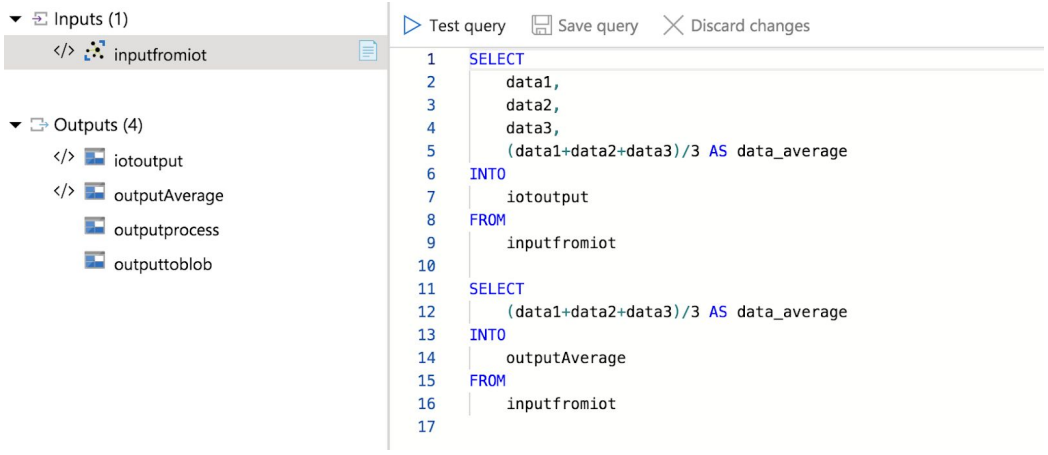
Create a storage account for storing the data.



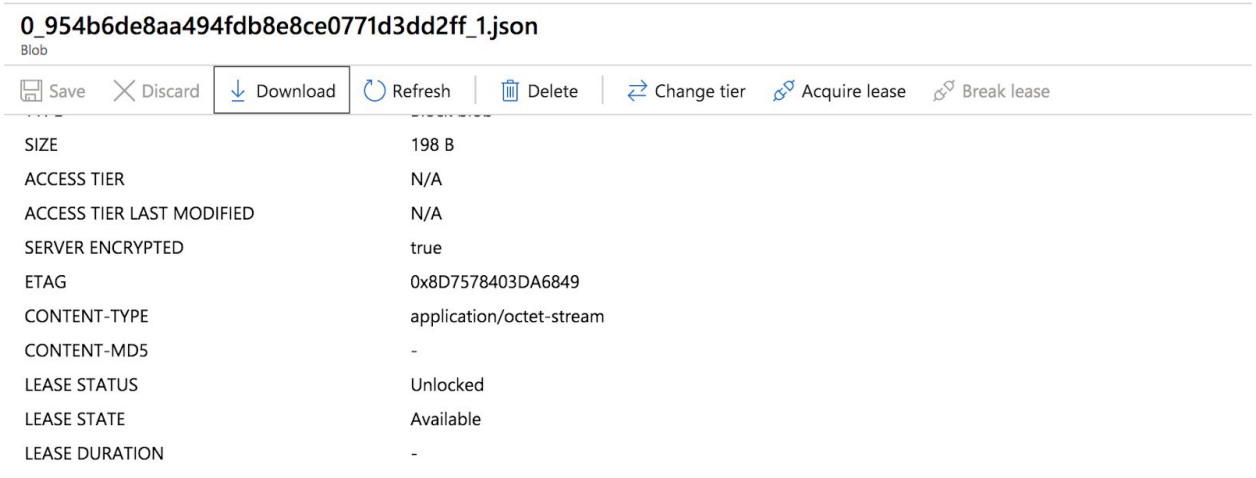
Add 2 blob storage output in the analytic job, one for store the raw data, and another for store processes data(get an average of the 3 data generated in this case). The set the output storing the data to the storage account just created.



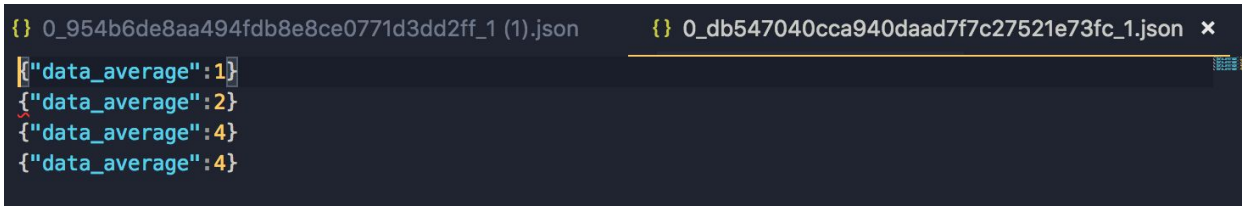
Edit Query doc, store the raw data to the blob, also, get the average of the data received and store the average as a newly processed data two another blob.

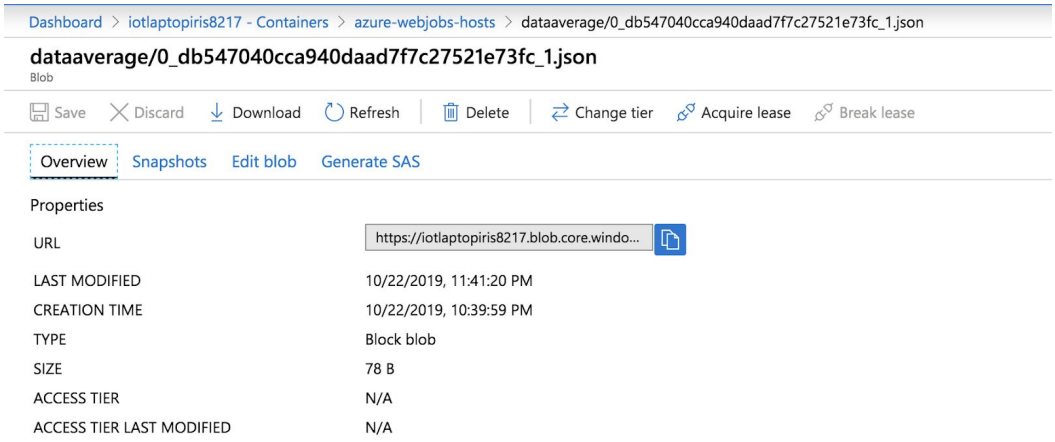


Get the raw data:



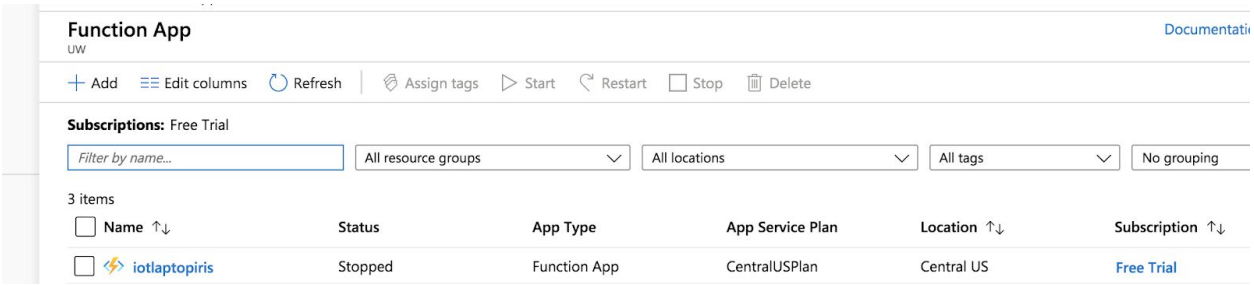
Get the processed data



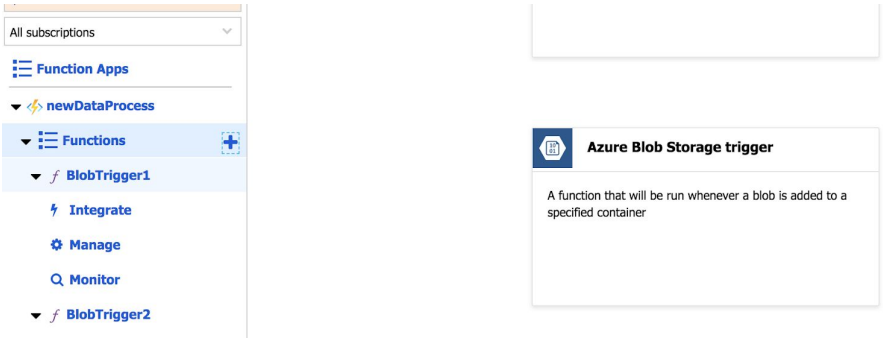


d). Connect Azure Functions to respond to Azure Blob events when new "processed data" is added. Your Azure Function may simply print out the content of newly added data, which is the "processed data" obtained in Step 3.

Create a new function app, set the correct language and consumer group (use node.js in this case)

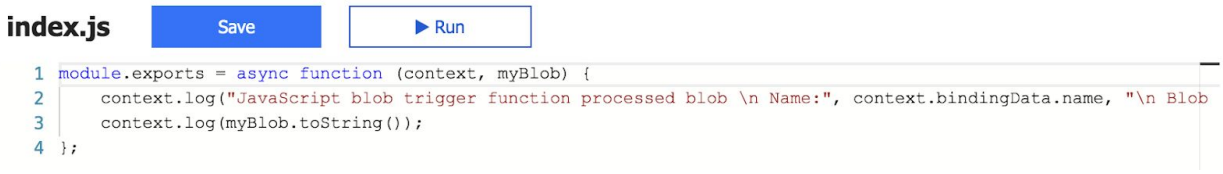


Create a new function, use existed azure blob storage trigger template here



Binding trigger variables on the portal page can also do it in the JSON file.

Run node.js code that prints the processed data whenever the blob receives a new message.




```
2019-10-23T07:20:45.640 [Information] Executing 'Functions.BlobTrigger2' (Reason='New blob detected: azure-webjobs-
hosts/dataaverage/0_e01f6bfa3acc4374b456802d7a38e2eb_1.json', Id=d0897dcc-94eb-405e-abb9-5e059fc7d48d)
2019-10-23T07:20:45.647 [Information] JavaScript blob trigger function processed blob
 Name: undefined
 Blob Size: 58 Bytes
2019-10-23T07:20:45.647 [Information] {"data_average":3}
{"data_average":5}
{"data_average":3}
```

```
2019-10-23T07:21:37.097 [Information] Executing 'Functions.BlobTrigger2' (Reason='New blob detected: azure-webjobs-
hosts/dataaverage/0_e01f6bfa3acc4374b456802d7a38e2eb_1.json', Id=c59f0aea-51a1-413a-9632-927acbeaf2e3)
2019-10-23T07:21:37.107 [Information] JavaScript blob trigger function processed blob
 Name: undefined
 Blob Size: 78 Bytes
2019-10-23T07:21:37.107 [Information] {"data_average":3}
{"data_average":5}
{"data_average":3}
{"data_average":4}
2019-10-23T07:21:37.108 [Information] Executed 'Functions.BlobTrigger2' (Succeeded, Id=c59f0aea-51a1-413a-9632-
927acbeaf2e3)
```

```
2019-10-23T07:22:28.343 [Information] JavaScript blob trigger function processed blob
 Name: undefined
 Blob Size: 98 Bytes
2019-10-23T07:22:28.343 [Information] {"data_average":3}
{"data_average":5}
{"data_average":3}
{"data_average":4}
{"data_average":4}
2019-10-23T07:22:28.344 [Information] Executed 'Functions.BlobTrigger2' (Succeeded, Id=181b2d69-2c36-4d44-a783-
6ce8bc1f29ae)
```

2 Problems Solving：

a). Select data in Query
Select * didn’t work for some reason, then I fixed it by selecting specific data, but still wanna know the reason.

b). Storage Limitation
Now in my blob, a new JSON file will be created automatically when the it reaches a specific size, didn’t find a place to do the setting.

c). Make deployment working on Visual Studio
I did the task4 on Visual Studio originally but met several problems, as .net doesn’t work well when deploying the function. Then I went back to the portal page and used the binding extensions provided on the portal, which works well.

3 Time Consuming
Around 8 hours